

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Currently amended) A method for testing changes in a software
2 program using a plurality of test cases, wherein the software program comprises a
3 first plurality of execution paths, the method comprising:
4 identifying one or more changed paths in the first plurality of execution
5 paths;
6 from the plurality of test cases, identifying one or more test cases that are
7 capable of executing the one or more changed paths by parsing and evaluating
8 names and parameters of one or more methods of a test case from the plurality of
9 test cases; and
10 executing the one or more of the identified test cases to test the changed
11 path.
- 1 2. (Original) The method of claim 1, wherein the software program
2 comprises one or more modules, and identifying one or more test cases comprises
3 identifying a changed module and determining whether the changed module
4 causes changes in the execution paths.
- 1 3. (Original) The method of claim 1, wherein identifying one or more
2 test cases comprises identifying a second plurality of execution paths in the
3 software program and determining the difference between the first and second
4 pluralities of execution paths.

- 1 4. (Original) The method of claim 3, wherein the difference
2 comprises at least one of a new path and a changed path.
- 1 5. (Cancelled)
- 1 6. (Cancelled)
- 1 7. (Currently amended) The method of claim 1, wherein identifying
2 one or more test cases comprises determining whether a test case ~~intersect~~
3 intersects one or more changed paths.
- 1 8. (Currently amended) The method of claim 7, wherein determining
2 whether a test case ~~intersect~~intersects one or more changed execution paths
3 comprises identifying a module of the software program included in both the test
4 case and a changed execution path.
- 1 9. (Original) The method of claim 8, wherein each module is
2 represented by a node number, and each execution path and test case is
3 represented by a string of node numbers, wherein identifying a module comprises
4 identifying a node number included in both a changed execution path and a test
5 case.
- 1 10. (Currently amended) A computer program product for testing a
2 software program using a plurality of test cases, the computer program product
3 comprising a computer usable medium having a computer readable program code
4 embodied thereon, the computer readable program code controlling the computer
5 to perform the operations of:

6 identifying one or more changed paths in a first plurality of execution
7 paths of the software program;
8 identifying one or more test cases that are capable of executing the one or
9 more changed paths by parsing and evaluating names and parameters of one or
10 more methods of a test case from the plurality of test cases; and
11 executing the identified one or more test cases to test the changed code of
12 the software program.

1 11. (Currently amended) The computer program product of claim 10,
2 wherein the software program comprises one or more modules, wherein
3 identifying one or more changed paths comprises identifying the changed module
4 and determining whether the changed module causes changes in the execution
5 paths.

1 12. (Currently amended) The computer program product of claim 10,
2 wherein identifying one or more changed paths comprises identifying a second
3 plurality of execution paths in the software program upon changing of the code
4 and determining the difference between the first and second pluralities of
5 execution paths.

1 13. (Original) The computer program product of claim 12, wherein the
2 difference comprises at least one of a new path and a changed path.

1 14. (Cancelled)

1 15. (Cancelled)

1 16. (Original) The computer program of claim 10, wherein identifying
2 one or more test cases comprises determining whether a test case intersects one or
3 more changed paths.

1 17. (Currently amended) The computer program of claim 16, wherein
2 determining whether a test case ~~intersect~~intersects one or more changed execution
3 paths comprises identifying a module of the software program included in both
4 the test case and a changed execution path.

1 18. (Original) The computer program of claim 17, wherein each
2 module is represented by a node number, and each execution path and test case is
3 represented by a string of node numbers, wherein identifying a module comprises
4 identifying a node number included in both a changed execution path and a test
5 case.

1 19. (Currently amended) A system for testing changes in a software
2 program using a plurality of test cases, wherein the software program comprises a
3 first plurality of execution paths, the system comprising:
4 means for identifying one or more changed paths in the first plurality of
5 execution paths;
6 means for identifying one or more test cases from the plurality of test cases
7 that are capable of executing the one or more changed paths by parsing and
8 evaluating names and parameters of one or more methods of a test case from the
9 plurality of test cases,
10 wherein the one or more identified test cases are executed to test the
11 changed code of the software program.

1 20. (Original) The system of claim 19, wherein the software program
2 comprises one or more modules, wherein upon changing of the code at least one
3 module is changed, and wherein identifying one or more test cases comprises
4 identifying the changed module and determining whether the changed module
5 causes changes in the execution paths.

1 21. (Original) The system of claim 19, wherein identifying one or more
2 test cases comprises identifying a second plurality of execution paths in the
3 software program upon changing of the code and determining the difference
4 between the first and second pluralities of execution paths.

1 22. (Original) The system of claim 21, wherein the difference
2 comprises at least one of a new path and a changed path.

1 23. (Cancelled)

1 24. (Cancelled)

1 25. (Original) The system of claim 19, wherein identifying one or more
2 test cases comprises determining whether a test case intersects one or more
3 changed paths.

1 26. (Currently amended) The system of claim 25, wherein determining
2 whether a test case ~~intersect~~intersects one or more changed execution paths
3 comprises identifying a module of the software program included in both the test
4 case and a changed execution path.

1 27. (Original) The system of claim 26, wherein each module is
2 represented by a node number, and each execution path and test case is
3 represented by a string of node numbers, wherein identifying a module comprises
4 identifying a node number included in both a changed execution path and a test
5 case.